NU-WRF 1.1 Software Description

Document Revision: Outline

1. Overview

NU-WRF 1.1 consists of a software package of coupled components, use case examples with data sets that exercise these components, and documentation on building the software and running the use cases.

2. WRF 3.2.1 and Related Packages

The NU-WRF 1.1 NWP model is based on WRF 3.2.1, which was released by NCAR in August 2010. Upgrading to the 3.2.1 release adds additional features to NU-WRF, including (from http://www.mmm.ucar.edu/wrf/users/):

- Bug fixes to various non-Goddard physics parameterizations
- New turbulence options
- Time-averaged flux output
- The WRF-Fire model
- Bug fixes and enhancements to the FDDA runtime options
- Vertical interpolation of relative humidity instead of specific humidity
- Fixes for digital filter initialization
- New 4D tracer array for chemistry and other applications
- New runtime option to skip variables from output files
- Bug fixes for nesting over complex terrain
- New option to increase vertical resolution by integer factor when nesting
- Improved domain decomposition for MPI and OpenMP

Our integration approach was to take NU-WRF 1.0 (based on WRF 3.1.1) and merge with WRF 3.2.1 using the Subversion revision control system. Source code conflicts identified by Subversion were resolved manually to produce a compilable source code. (This procedure was repeated for other software described below.) The resulting executable is being evaluated with a series of regression tests based on existing use cases.

The WPS preprocessing software was upgraded to version 3.2.1. This software is used to process GRIB, GRIB2, and static terrestrial data for input to NU-WRF. Features introduced by WPS 3.2.1 (as documented at http://www.mmm.ucar.edu/wrf/users/) include:

- Support for processing UK Met Office and NOGAPS fields
- Conversion of ECMWF snow depth to appropriate units
- Bug fix to processing ECMWF relative humidity
- Improved handling of missing relative humidity values in the upper atmosphere
- Upgrades to 2008 revisions of GRIB Table 3.2
- Several fixes to projection and interpolation options in metgrid
- Fixed interpolation of Gaussian grid data
- Added support for AFWA ice input

The WPP post-processing software was upgraded to version 3.2. This program allows conversion of NU-WRF netCDF output to GRIB format, along with calculation of a number of diagnostic variables. WPP is a front-end to running the MET verification software on NU-WRF simulations. Changes introduced by WPP 3.2 (as documented at http://www.dtcenter.org/wrf-nmm/users/) include:

- Bug fix to radar reflectivity computation using the Ferrier scheme
- Bug fix to interpolate 10-m winds from mass points to velocity points
- Bug fix to calculated dewpoint values
- Bug fix to calculated temperatures
- Enhancements to accommodate regular latitude/longitude projection

The MET verification software was upgraded to version 3.0. This software incorporates a number of verification measures appropriate for gridded and point-based data. Changes introduced by MET 3.0 (as documented at http://www.dtcenter.org/met/users/) include:

- Support for verifying ensemble forecasts
- Multi-category contingency tables and skill scores
- Plotting of point observations
- Support for Air Force WWMCA cloud analysis
- Various bug fixes

The ARWpost post-processor was upgraded to version 2.2. This program allows conversion of NU-WRF netCDF output to GrADS format (and optionally Vis5d), along with calculation of a number of diagnostic variables. Changes introduced by ARWpost 2.2 (documented in the ARWpost README file) include:

- Fix for calculating relative humidity
- Using snow and graupel when calculating radar reflectivity
- Added latitude/longitude projection
- Added an extrapolation option

The RIP post-processing software was upgraded to version 4.6. This program allows for plotting of NU-WRF netCDF output using NCAR Graphics, along with calculation of a number of diagnostic variables. The only significant change was a bug fix to calculating geopotential height (personal correspondence from NCAR).

The GEOS2WRF/MERRA2WRF pre-processor software was upgraded to version 1.0.1-beta2. These programs process GEOS-5 and MERRA atmospheric data (respectively) into WPS intermediate format for subsequent use by WPS and NU-WRF. The upgrade incorporates a bug fix for error handling when a bad PBS charge code is used on DISCOVER.

The GOCART2WRF pre-processor program was upgraded to incorporate several bug fixes. The program is responsible for processing GOCART netCDF files for use as aerosol boundary conditions for NU-WRF.

3. Build-All Mechanism

The WRF modeling system is composed of a number of software packages, each of them with their own separate build system. To make it easier for the user to create desired executables and to more easily resolve dependencies between packages, the NU-WRF 1.1 includes a set of high-level “wrapper” scripts for building. With this new system, the user can build executables using a single script: build.sh in the top-level directory. This build.sh reads in a separate configure script (default is discover.cfg) defining paths to libraries and include files, as well as the appropriate module environment for compilers and MPI implementation. The user may optionally override the default configuration file with the command line flag “--config” followed by the file name. The script requires as least one other command line argument specifying what to build:

- `./build.sh wrf  # Build executables in WRFV3 directory`
- `./build.sh chem # Same as wrf option, but compiles with chemistry`
- `./build.sh wps # Build executables in WPS directory`
- `./build.sh wpp # Build executables in WPPV3 directory`
- `./build.sh rip # Build executables in RIP4 directory`
- `./build.sh arwpost # Build executable in ARWpost directory`
- `./build.sh geos2wrf # Build executables in utils/geos2wrf directory`
- `./build.sh merra2wrf # Same as geos2wrf option`
- `./build.sh gocart2wrf # Build executable in utils/gocart2wrf directory`
- `./build.sh sst2wrf # Build executable in utils/sst2wrf directory`
- `./build.sh met # Build executables in MET directory`
- `./build.sh all  # Build all executables without chemistry`
- `./build.sh allchem # Build all executables with chemistry`

The script also accepts multiple arguments allowing users to pick and choose executables (e.g., ‘./build.sh wps wrf wpp met’ will build executables in the four directories).

One complication addressed by the build system is that several packages (WPS, WPPV3, and ARWpost) are dependent on libraries and object files from WRFV3. In addition, WPPV3 requires WRFV3 to be built in serial mode. To account for these dependencies, the ./build.sh has the following behavior:

- WRFV3 will always be built with MPI, except when WPPV3 needs to be built.
- If WPPV3 needs to be built, WRFV3 will first be built in serial mode, then WPPV3 will be built, and then WRFV3 will immediately be cleaned. This procedure will always occur before any other executables are built.
- If building WPS or ARWpost, the required WRFV3 files are first checked to see if they exist. If they are missing, WRFV3 will be compiled prior to WPS and/or ARWpost, regardless of whether ‘wrf’ is specified at the command line.

An additional complication is specific to NU-WRF: The coupling of LIS to WRF introduces several new library dependencies on WPPV3, including ESMF 3.1.0rp3. Furthermore, since WPPV3 expects WRFV3 to be compiled serially, the ESMF library must likewise be built in serial mode. To account for this, two separate ESMF libraries are specified in discover.cfg, one for each compilation mode. Also, the template configuration for WPPV3 is modified to add LIS-specific libraries so all necessary routines are resolved.

The top-level build.sh calls lower-level build.sh wrapper scripts located in each package directory (WRFV3, WPS, etc.). Configuration settings are passed to the lower-level scripts via environment variables. Each lower-level script is customized to directly manage the appropriate build system (e.g., ./configure and ./compile for WRFV3, make for utils/geos2wrf), and to inject the appropriate configuration settings to the existing build system. For example, the build.sh for WPS will modify the configure.wps file generated by ./configure to update several library paths; the modified configure.wps is then used by the ./compile script.

Currently the build system is implemented for the DISCOVER supercomputing cluster using Intel compilers. It may be desirable to add support for other computing systems (e.g., at

NAS), so other configuration files could be written as well (e.g., pleiades.cfg). However, more significant changes would be required to specify alternative compilers or compiler flags and pass this information to the different packages.